RCNN-CTC Model for Handwritten Text Recognition: In-depth Analysis of the IAM Dataset

Sai Thikekar, Anurag Shirsekar, Yash Chhaproo, Uzair Shaikh, Lifna C S

Department of Computer Engineering, Vivekanand Education Society's Institute Of Technology, Mumbai, India

Abstract— **Due to the ease of using a pen tip instead of a keyboard, the majority of scripts are currently handwritten; as a result, errors are frequent because human handwriting is illegible. Recognition of handwriting is crucial to avoiding this issue. Handwritten offline The need to remove errors resulting from misreading handwritten writing and the need for automation to increase productivity have made text recognition (OHTR) one of the main research topics in recent years. This technology is used in a number of different sectors, including signature verification, postal address recognition, and handwritten application interpretation. In this study, Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), which employs the architecture of Recurrent Neural Network (RNN), and Connectionist Temporal Classification (CTC) are used to accomplish offline handwritten text recognition. The IAM database, which contains handwritten English text, is used to train and test the neural network. This work is implemented utilizing image segmentation-based handwritten text recognition, where TensorFlow is utilized for text recognition training and OpenCV is used for image processing. Python was used to develop the entire system, and a word document was used to display the results.**

Keywords— **Handwritten Text Recognition, IAM dataset, RCNN-CTC, deep learning.**

I. INTRODUCTION

The study of identifying handwritten text in both human and computer-generated fonts is known as hand text recognition. As technology developed, there was a strong desire to merge analog records into digital ones in order to reduce duplication and the communication chain. A handwritten text recognition system is in high demand as the world moves closer to complete digitization. The difficulty in putting this approach into practice, nevertheless, comes from the variety of handwritten words' features, including rounded and slanted characters, diacritical dots, crossbars, and humped letters. In order to determine which words are most likely, a decent handwriting recognition system needs to correctly detect the distorted characters.

To improve system efficiency, this application uses the IAM Dataset, which contains over 100,000 photos of unconstrained handwritten text, for system validation, testing, and training. This dataset is used to train the neural network so that it can eventually detect handwriting. Recurrent neural networks (RNNs) have less processing power, but they can handle greater input. Conversely, more data is required for the Convolutional Neural Network (CNN) to be trained. An adaptive strategy by incorporating both is proposed for offline HTR in this handwritten text recognition system. Here, CNN and RNN are used in succession to train the dataset. To model the probability of a label, a connectionist temporal classification (CTC) network is fitted in conjunction with an RNN through training. The user's handwritten text is interpreted by this technology as visuals. These photos are pre-processed; the goal of this step is to reduce the amount of data, remove flaws, and normalize the images in order to create a set of relevant data that can be used to segment the images.

Three steps are involved in the implementation of this project: text recognition, word segmentation, and line segmentation. A crucial step in handwritten text identification is line segmentation, which starts with greyscale picture conversion, moves on to obtaining inverted binary images, dilates those images, and ends with boundary boxes drawn over each line. Scale-space segmentation is used in word processing. In order for the recognition system to recognize the text, this procedure splits the text lines that are produced as the output of line segmentation into discrete words.

Our planned research project aims to analyze the difficulties in handwritten text recognition using CNN and RNN, and to compute the loss using the Connectionist temporal classification (CTC) network. The other goals are to evaluate the CNN RNN method's performance in comparison to traditional methods and to develop a system that can expedite processing, save time, and lower the likelihood of errors. Using the sophisticated automation mechanisms found in handwritten text recognition systems improves man-machine interactions in many areas.

In this study, we develop a CNN, RNN, and CTC layer model. The input image is given to the CNN layers. These layers are trained to recognize relevant features in the image. There are three processes in each stratum. The convolution procedure is followed by the non-linear RELU function. In the end, a pooling layer produces a smaller input image by condensing image regions. In the feature sequence, there are 256 features for each time-step. Through the RNN, pertinent information is transmitted in this order. This makes use of the well-known RNN implementation for Long Short-Term Memory (LSTM). Finally, the CTC receives the RNN output matrix used to decode the output text.

The following summarizes the paper's primary contribution:

- To get the best outcomes, the author suggests a unique approach that combines the CNN and RNN networks.
- Using OpenCV contour algorithms, the text paragraph images are converted into word images, which are then supplied into the network model for recognition.
- The IAM word image dataset is used to train the model. The overall structure of the paper is as follows: Section 2 addresses

the associated works. In Section 3, the suggested methodology is covered. The results are discussed in Section 4, and the conclusion is covered in Section 6.

II. LITERATURE REVIEW

Although there are numerous approaches for handwritten text identification, each approach has drawbacks since handwritten writing is not straightforward. For this reason, the most efficient approach with a straightforward technique is applied.

The goal of Shivakumara et al. [1] is to identify license plates on cars in Malaysia that have both a white and a black backdrop. They employ a CNN-RNN based recognition system for feature extraction and a BLSTM-based system for context-aware information extraction (classification) in order to solve this difficulty. MIMO and UCSD are the datasets they used in their investigation. Classification is an essential component of this system since all of the current techniques for recognizing license plates are ineffective for numerous unfavorable circumstances. License plate recognition is accomplished by LSTM. Their work thus demonstrates the value of classification in enhancing recognition performance.

The technique of fusing sequence-to-sequence networks—also known as encoder-decoder networks—with deep neural networks is employed by Sueiras et al. [2]. In order to recognize a given word, the proposed architecture seeks to identify characters and link them with their neighbors. These datasets, which include numerous handwritten texts on white backgrounds, are used for training and testing IAM and RIMES. In the test set, the error rates are 6.6% in RIMES and 12.7% in IAM. Compared to handwriting recognition, this technology is more effective for language translation and speech-to-text conversion.

Levenbreg and Firefly were utilized by Sampath and Gomathi [3].For optical character recognition, the Levenbreg-Marquardt and Firefly algorithms are merged to create a hybrid neural network approach called the Marquardt (FLM) algorithm. The system's speed and accuracy are increased by this hybrid neural network approach, which combines the benefits of both algorithms. To demonstrate the effectiveness of the hybrid method as it relates to gradient feature descriptors, FLM with feed-forward is contrasted with an SVM-based technique. Their system's drawback is that it requires more intricate architecture to carry out basic functions.

Line detection in handwritten texts has proven to be a significant challenge for processing scanned documents, as discussed by Vo et al. [4]. The majority of current methods use hand-drawn characteristics or heuristic techniques to estimate text line locations. A unique method has been put forth that involves first training a fully convolutional network (FCN) to anticipate the text line structure in the document's scanned images. To guarantee that the segmentation procedure is completed, the touching characters have been separated and assigned to the various text lines using the line adjacency graph. The ICDAR2013 Handwritten Segmentation data, which demonstrated excellent performance for the combination of several languages and multi-skewed text lines, was used to evaluate the system's robustness.

III. PROPOSED METHODOLOGY

As illustrated in Fig.1, the block diagram serves as a representation of the proposed work's outline. Training the dataset is the initial step in the procedure. CNN and RNN layers are utilized in the dataset's training. To obtain the trained model, the output is obtained and the ground truth text is sent through the CTC layer. The text in the input image is then recognized using the trained model that has been obtained.

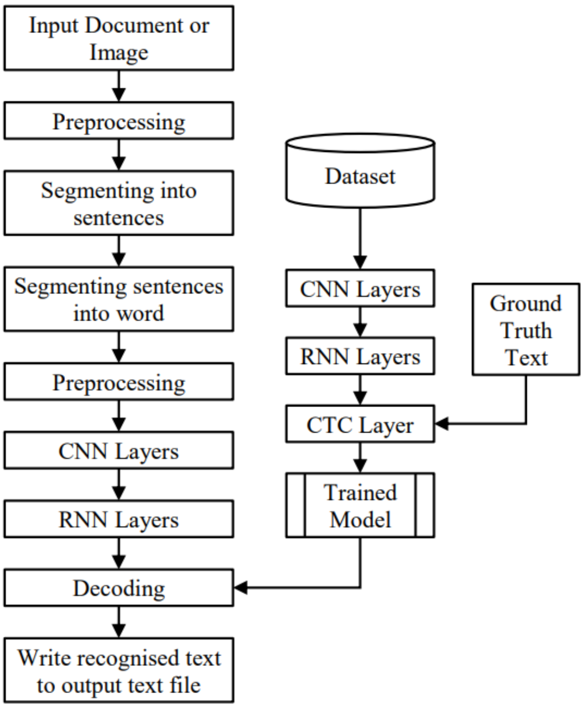


Fig.1. Block Diagram

The resolution of the input handwritten image is changed as part of the preprocessing step. Dividing the paragraph image into line images is the initial stage in the identification process. Word images are then created by further segmenting line images. Following preprocessing, the word pictures are run through the same CNN and RNN layers that were employed during training. The CTC layer, or decoding level, receives the output from the RNN layers and uses the trained model to decode the output text.

It has been successfully developed to extract word pictures from a paragraph and combine CNN, RNN, and CTC approaches to train the NN model. An end-to-end system for recognising handwritten text has been developed and put into practise.

3.1 CLASSIFIERS

3.1.1 Convolutional Neural Network:

Convolutional neural networks are among the most significant algorithms in deep learning (CNN). The process begins with an input image, extracts features from the image, and then separates one feature from another. The relationships between the neurons in the human brain serve as its inspiration. The feature detection layer's learning is powered by the training data that CNN uses internally [7].

3.1.2 Recurrent Neural Network:

Recurrent neural networks (RNNs) are another popular deep learning algorithm for the sequential processing of input in applications like text and speech recognition. Recurrent refers to the fact that the network does the same operation repeatedly for each element in the data sequence, with the resulting output being determined by the values of the network's preceding outputs [6].

3.1.3 Connectionist Temporal Classification:

Through the application of a novel differentiable cost function, Connectionist Temporal Classification (CTC) directly trains RNNs to identify and categorize the unsegmented sequences. An extra blank symbol is added to the list of potential labels that the recurrent neural networks can produce in order to employ this cost function. The RNN's output layer represents probabilities over all potential labels.

3.2 PROPOSED CNN-RNN MODEL

The initial stage of the neural network (NN) processing involves feeding the input image into CNN layers, which are specifically trained to extract essential features from the image. Each layer conducts three operations: convolution, activation through Rectified Linear Unit (ReLU), and downsizing the image via pooling to identify distinct image regions. Convolution applies a 5x5 filter kernel for the first two layers and a 3x3 filter kernel for the subsequent three layers, followed by ReLU activation. This process results in a downsized version of the image, with a height reduction of 2 and the addition of channels through feature mapping, ultimately generating a 32x256 output sequence.

The subsequent step involves processing this sequence through an RNN, specifically implemented using Long Short-Term Memory (LSTM) networks for their capacity to handle long-range dependencies and superior training characteristics compared to Vanilla RNNs. Each time step of the sequence comprises 256 features, which are applied to the LSTM. The LSTM output sequence is then mapped to a 32x80 matrix, accommodating the 79 different characters present in the IAM dataset, along with an additional character reserved for generating blank labels in the CTC operation, thereby ensuring each step contains 80 elements.

During NN training, the ground truth text and the RNN output matrix are fed into the Connectionist Temporal Classification (CTC) layer. This layer decodes the output matrix into text, compares it with the ground truth text, and generates a loss value. The recognized text length is fixed at 32 characters. The average of these loss values is utilized to train the NN, facilitated by an RMSProp optimizer. Once trained, the model is deployed to recognize input images, achieving a word error rate of 10.62% upon evaluation.

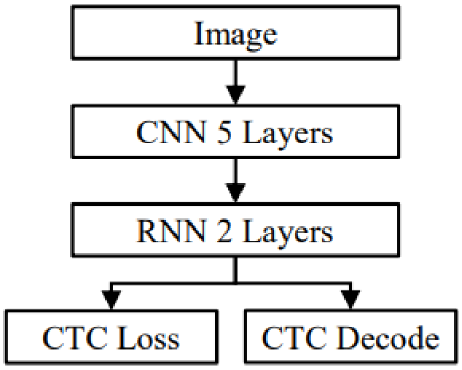


Fig.2. Overview of the model

3.3 IMPLEMENTATION

This system introduces an adaptive approach for offline Handwritten Text Recognition (HTR) by combining both Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) techniques. The IAM dataset, comprising around 100,000 words in British English, is sequentially trained with both neural network architectures.

3.3.1 Pre-processing

The pre-processing phase of handwriting involves a sequence of operations conducted on handwritten text data before applying recognition algorithms. This stage aims to reduce dimensionality, eliminate inconsistencies, and normalize the data, resulting in a dataset more conducive to image-based data segmentation. Pre-processing is executed through two primary steps: Line segmentation and Word segmentation, leveraging the OpenCV library.

Line segmentation entails converting the input image to grayscale, followed by transforming it into an inverse binary image. Subsequent dilation of this binary image and identification of its contours, which are then enclosed within bounding boxes, yield separate line images.

In the word segmentation process, the text lines obtained from line segmentation undergo segmentation into individual words. The line image is pre-processed, followed by the application of a filter kernel. The resultant filtered image is dilated to generate an inverted binary image. Contours of this image are identified, and bounding boxes are employed to encapsulate these contours, isolating individual words. These segmented words are then stored as separate word images.

The pre-processing stage for handwritten text involves critical steps aimed at enhancing the data for subsequent recognition algorithms. It encompasses both line segmentation and word segmentation processes. In line segmentation, operations such as grayscale conversion, binary image transformation, dilation, contour identification, and bounding box application yield distinct line images. Similarly, word segmentation involves similar steps applied to text lines, resulting in the isolation and storage of individual word images.

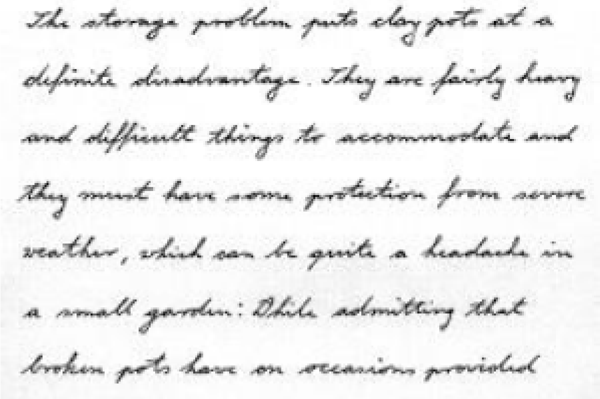

The storage problem puts clay pots at a
definite disadvantage. They are fairly heavy
and difficult things to accommodate and
they must have some protection from severe
weather, which can be quite a headache in
a small garden: While admitting that
broken pots have on occasion provided

Fig.3. Gray scale image

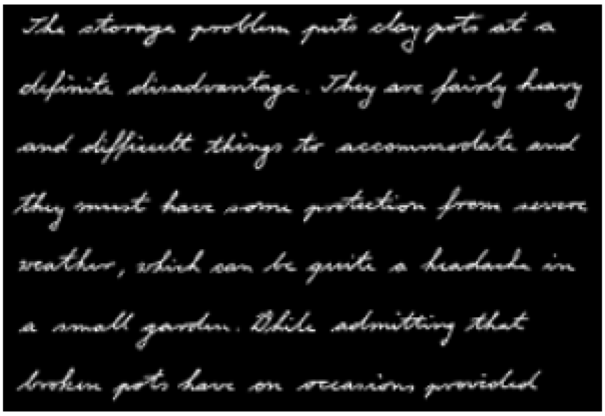

The storage problem puts clay pots at a
definite disadvantage. They are fairly heavy
and difficult things to accommodate and
they must have some protection from severe
weather, which can be quite a headache in
a small garden. While admitting that
broken pots have on occasion provided

Fig.4. Inverted binary image

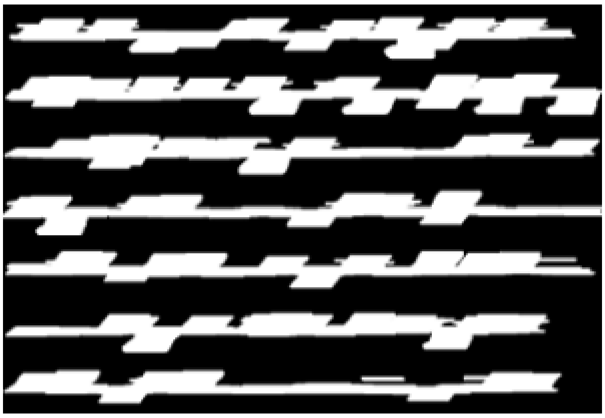


Fig.5. Dilated image

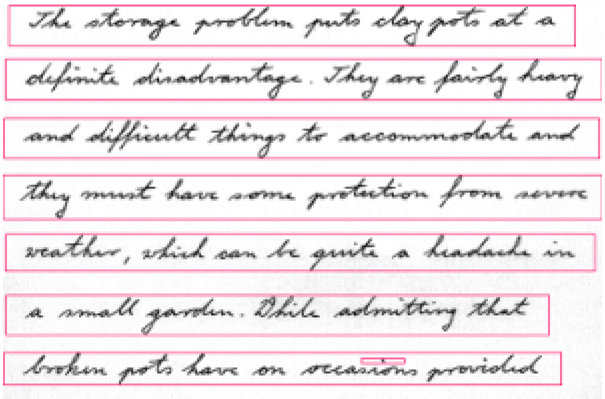

The storage problem puts clay pots at a
definite disadvantage. They are fairly heavy
and difficult things to accommodate and
they must have some protection from severe
weather, which can be quite a headache in
a small garden. While admitting that
broken pots have on occasion provided

Fig.6.Bounding boxes over lines

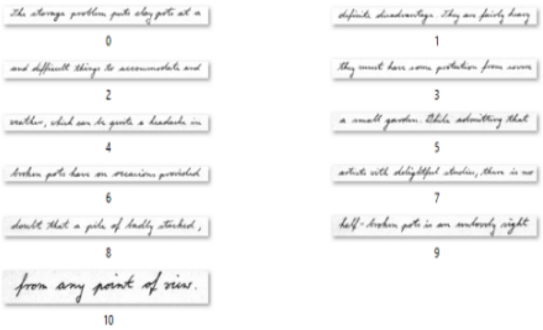

The storage problem puts clay pots at a
0
definite disadvantage. They are fairly heavy
1
and difficult things to accommodate and
2
they must have some protection from severe
3
weather, which can be quite a headache in
4
a small garden. While admitting that
5
broken pots have on occasion provided
6
artists with delightful studies, there is no
7
doubt that a pile of badly stacked,
8
half-broken pots is an unlovely sight
9
from any point of view.
10

Fig.7. Line images

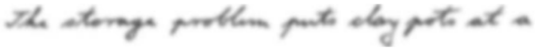


Fig.8. Filtered line image

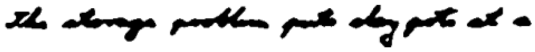


Fig.9. Dilated line image

Fig.10. Inverted line image

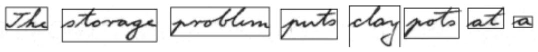

The
storage
problem
puts
clay
pots
at
a

Fig.11. Bounding boxes over words

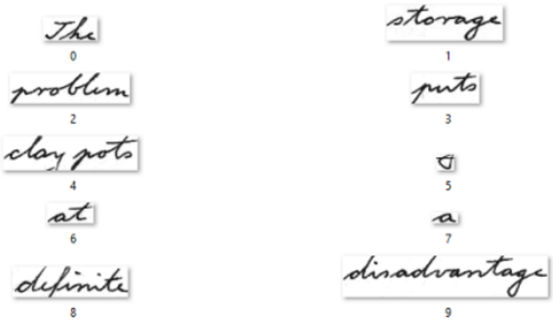

The
0
storage
1
problem
2
puts
3
clay pots
4
5
at
6
a
7
definite
8
disadvantage
9

Fig.12.Word images

3.3.2 Dataset

The training and validation of the model are conducted using the IAM dataset, comprising over 100,000 handwritten word images, resulting in enhanced algorithm efficiency. Two different ratios, 65:35 and 50:50, are employed for training and validation, with the latter yielding a higher average probability. The trained model achieves a word error rate of 10.62%.

Table.1. Specification of IAM dataset

Description	Count
Number of writers contributed samples of their handwriting	657
Number of pages of scanned text	1539
Number of isolated and labelled sentences	5685
Number of isolated and labelled text lines	13353
Number of isolated and labelled words	115320

Table.2. Average probability for two sets of Training-Validation ratio

Image	50%-50%	65%-35%
1	74	72
2	54	45
3	70	68
4	55	57
5	80	75
6	63	66
7	49	51
8	46	45
9	70	68
10	72	70
11	78	75
12	76	75
13	71	71
14	79	75
15	71	69
Average	67.2	65.47

IV. RESULTS AND DISCUSSION

The offline handwritten character recognition system is meticulously designed to enhance its performance at every stage of the process, spanning from pre-processing and training to eventual recognition.

4.1 PRE-PROCESSING

During pre-processing, paragraphs are initially converted into lines and subsequently into words. This involves converting the image to grayscale and extracting words, a task efficiently accomplished using OpenCV. Such pre-processing significantly aids in enhancing text recognition efficiency.

4.2 TRAINING AND VALIDATION

The training and validation of the model are conducted using the IAM dataset, comprising over 100,000 handwritten word images, resulting in enhanced algorithm efficiency. Two different ratios, 65:35 and 50:50, are employed for training and validation, with the latter yielding a higher average probability. The trained model achieves a word error rate of 10.62%.

4.3 TEXT RECOGNITION

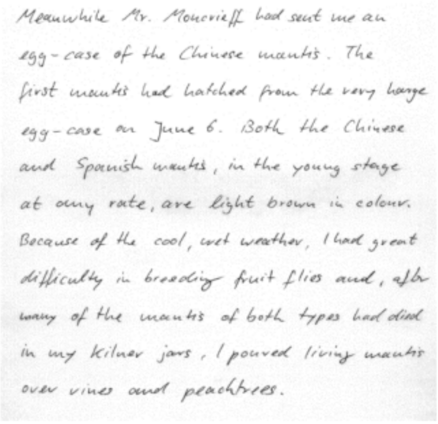
During the assessment of the proposed system's performance, the probability of each identified word falls within the range of 50% to 98% for images sourced from the IAM dataset, and between 45% to 90% for custom handwritten images. Similarly, the probability of the recognized paragraph ranges from 70% to 80% for IAM dataset images, whereas for custom images, it lies within the range of 50% to 70%, as depicted in Table 3

Table.3. Probability results of word and paragraph images

Probability	IAM Dataset Image	Custom Handwritten Image
Average Probability of all words in a paragraph	70% to 80%	50% to 70%
Individual probability of word image in a paragraph	50% to 98%	45% to 90%

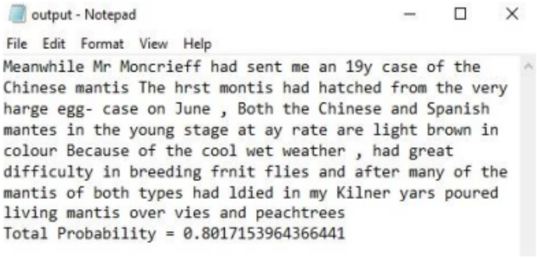
4.4 OUTPUT

The input paragraph images, sourced from both the IAM dataset and custom handwritten samples, undergo processing through various steps outlined in the model. Subsequently, the resulting output text is saved as a text file, accompanied by the average probability of all words within the paragraph. Fig. 13 illustrates an input image from the IAM dataset, with its corresponding recognized output and total probability depicted in Fig. 14. It's crucial to note that the method used to generate these images significantly impacts accuracy, as it influences the quality of input images fed into the system. Images captured by cameras typically exhibit lower quality compared to those scanned by dedicated scanners. Factors such as inadequate lighting, shadows, glares, blurring, degradation in corners, skewness, tilting, and aspect ratio discrepancies can all contribute to issues affecting image quality.



Meanwhile Mr. Moncrieff had sent me an
egg-case of the Chinese mantis. The
first mantis had hatched from the very large
egg-case on June 6. Both the Chinese
and Spanish mantis, in the young stage
at any rate, are light brown in colour.
Because of the cool, wet weather, I had great
difficulty in breeding fruit flies and, after
many of the mantis of both types had died
in my Kilner jars, I poured living mantis
over vines and peachtrees.

Fig.13. Input image (IAM dataset)



```
Meanwhile Mr Moncrieff had sent me an 19y case of the
Chinese mantis The hrst montis had hatched from the very
harge egg- case on June , Both the Chinese and Spanish
mantes in the young stage at ay rate are light brown in
colour Because of the cool wet weather , had great
difficulty in breeding frnit flies and after many of the
mantis of both types had ldied in my Kilner yars poured
living mantis over vies and peachtrees
Total Probability = 0.8017153964366441
```

Fig.14. System Output

V. CONCLUSION AND FUTURE WORK

This system introduces an adaptive approach to offline paragraph recognition, employing a sequential processing pipeline involving both CNN and RNN. Initially, input paragraph images undergo pre-processing utilizing OpenCV contour techniques, where they are segmented into line images. These line images are further processed into word images, which serve as input to the NN model layers during recognition. The output of CNN layers is subsequently processed by RNN layers, and the resulting output is decoded by the Connectionist Temporal Classification (CTC) layer to yield the recognized text. The consecutive use of CNN and RNN demonstrates a progressive improvement in accuracy.

Future enhancements to this system are envisioned, including the incorporation of hybrid datasets and experimentation with different activation functions. Additionally, plans involve increasing the number of neural network layers for further refinement. The system's capabilities are intended to be extended to include online recognition and support for different languages. Furthermore, efforts will be directed towards enabling the system to recognize degraded text or fragmented characters, thereby enhancing its robustness and applicability across diverse scenarios.

VI. REFERENCES

[1] P. Shivakumara, D. Tang, M. Asadzadehkaljahi, T. Lu, U. Pal and M. Hossein Anisi, "CNN-RNN based Method for License Plate Recognition", CAAI Transactions on Intelligence Technology, Vol. 3, No. 3, pp. 169-175, 2018.

[2] J. Sueiras, V. Ruiz, A. Sanchez and J.F. Velez, "Offline Continuous Handwriting Recognition using Sequence to Sequence Neural Networks", Neurocomputing, Vol. 289,pp. 119-128, 2018. ISSN: 2229-6956 (ONLINE) ICTACT JOURNAL ON SOFT COMPUTING, OCTOBER 2021, VOLUME: 12, ISSUE: 01 2463

[3] A. Sampath and N. Gomathi, "Handwritten Optical Character Recognition by Hybrid Neural Network Training Algorithm", The Imaging Science Journal, Vol. 67, No. 7, pp. 359-373, 2019.

[4] Q. Vo, S. Kim, H. Yang and G. Lee, "Text Line Segmentation using a Fully Convolutional Network in Handwritten Document Images", IET Image Processing, Vol. 12, No. 3, pp. 438-446, 2018.

[5] J. Chung and T. Delteil, "A Computationally Efficient Pipeline Approach to Full Page Offline Handwritten Text Recognition", Proceedings of International Conference on Document Analysis and Recognition Workshops, pp. 1-13, 2019.

[6] Y. Chherawala, P. Roy and M. Cheriet, "Feature Set Evaluation for Offline Handwriting Recognition Systems: Application to the Recurrent Neural Network Model", IEEE Transactions on Cybernetics, Vol. 46, No. 12, pp. 2825-2836, 2016.

[7] Y. Weng and C. Xia, "A New Deep Learning-Based Handwritten Character Recognition System on Mobile Computing Devices", Mobile Networks and Applications, Vol. 25, pp. 1-22, 2019.

[8] A. De Sousa Neto, B. Bezerra, A. Toselli and E. Lima, "HTR-Flor: A Deep Learning System for Offline Handwritten Text Recognition", Proceedings of International Conference on Graphics, Patterns and Images, pp. 1-8, 2020.